AGI logic tutorial

YKY

February 17, 2021

Summary

AGI univeral logic

Contents

0 Background

optimization problem,

$$\text{maximize: } \int_0^\infty R\,dt \tag{0.0.1}$$

where $R(t)$ = reward at time t. $\int_0^\infty$ **time horizon**.

N (0.0.1)

Architecturally, the AI is a **dynamical system** that constantly updates its "state" $\boldsymbol{x}$ via: [*]

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{0.0.2}$$

$$\boldsymbol{x}_{t+1} = \boldsymbol{F}(\boldsymbol{x}_t) \tag{0.0.3}$$

$\boldsymbol{F}$ transition function.

$$\begin{tikzcd} \boldsymbol{x} \arrow[loop above, "\boldsymbol{F}"] \end{tikzcd} \tag{0.0.4}$$

Our goal is to **learn** the function $\boldsymbol{F}$, implemented as a **deep neural network**. $\boldsymbol{F}$

[*] Part of the state $\boldsymbol{x}$ contains **sensory input** and **action output** that allow the AI to interact with the external environment.

1 Structure of logic

The central tenet of my theory is that the state $\boldsymbol{x}$ of the AI system is consisted of **logic propositions** and that $\boldsymbol{F}$ plays the role of the **logic consequence** operator $\vdash$:

$$\boxed{\text{propositions}} \;\vdash\!\xrightarrow{\boldsymbol{F}}\; \boxed{\text{propositions}} \tag{1.0.1}$$

So our goal now is to elucidate the structure of $\vdash$. Currently the most elegant formulation is given by **categorical logic** or **topos theory**.

“synthesize” ideas ideas

AGI Curry-Howard isomorphism....

2 Curry-Howard correspondence

Curry-Howard isomorphism

$$\begin{array}{ccc}
\text{(syntax)} & A \Rightarrow B & \\
& \boxed{\text{logic}} & A \Longrightarrow B \\
\hdashline
& \boxed{\text{program}} & \blacksquare \xmapsto{f} \blacksquare \\
\text{(proof)} & A \quad \text{map} \quad B & \text{“proof witness”}
\end{array} \tag{2.0.1}$$

$\blacksquare$.

$f(x) = x + 2$

$$\begin{array}{c}
f : \mathbb{R} \longrightarrow \mathbb{R} \\
\hdashline
x \longmapsto x + 2
\end{array} \tag{2.0.2}$$

$$x \quad \Longrightarrow x + 2 \tag{2.0.3}$$

x (witness)

$$\begin{array}{c}
\boxed{x \quad} \\
\hdashline
\blacksquare
\end{array} \tag{2.0.4}$$

This is called the **Brouwer-Heyting-Kolmogorov (BHK) interpretation**.
subtle Brouwer-Heyting-Kolmogorov-Schönfinkel-Curry-Meredith-Kleene-Feys-Gödel-Läuchli-Kreisel-Tait-Lawvere-Howard-de Bruijn-Scott-Martin-Löf-Girard-Reynolds-Stenlund-Constable-Coquand-Huet-Lambek ....

HoTT (homotopy type theory)
$\in [0,1]$ fuzzy fuzzy §7

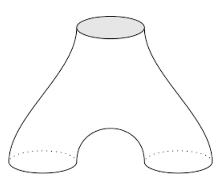
John Baez (1961-)

Curry-Howard isomorphism states A transitions $\overset{f}{\mapsto}$

logic	**computation**	**category theory**	**physics**	**topology**
proposition	type	object	system	manifold
proof	term	morphism	process	cobordism

(2.0.5)

Curry-Howard Lambek John Baez & M. Stay *Physics, Topology, Logic and Computation: a Rosetta stone* [2010]. physics Hilbert space operators topology cobordism "pair of pants"



(2.0.6)

In string theory strings string

2.1 Type theory

program **computation** type theory.

```
define length(s: String): Integer = { .... }
```
(2.1.1)

length() String Integer.

$$f : A \to B \tag{2.1.2}$$

type theory

$$\overbrace{t}^{\text{term}} : \overbrace{T}^{\text{type}} \tag{2.1.3}$$

notation $t : T$ $t \in T$

types terms

type theory type **context**

$$\overbrace{x : A}^{\text{context}} \vdash \overbrace{f(x) : B}^{\text{type assignment}} \tag{2.1.4}$$

program "declare" program

$\vdash$ **type assignment** type theory

λ-calculus

program λ-calculus

λ-calculus

$$f(x) \triangleq x^2 \tag{2.1.5}$$

λ-

$$f \triangleq \lambda x.\ x^2 \tag{2.1.6}$$

λ- f

λ-calculus Alonso Church **substitution** Substitute

Church λ-calculus **Turing machines** AI John McCarthy λ-calculus **Lisp** functional programming language

Curry-Howard correspondence

Curry-Howard type A A type A A (proof, or proof witness)

$A \Rightarrow B$ the function type $A \to B$ B^A type
$f : A \to B.$ type $A \to B$ (inhabited) $A \Rightarrow B$

2.2 Intuitionistic logic

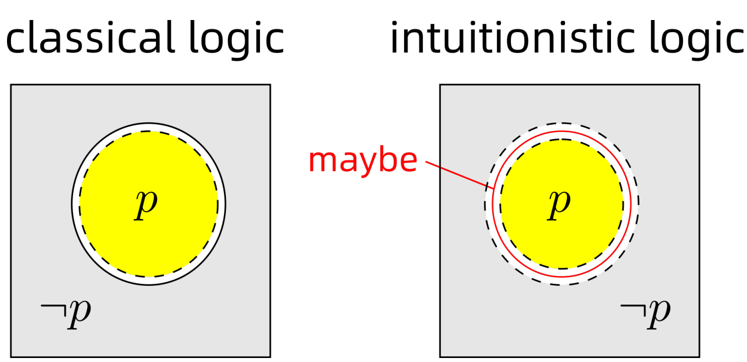
Curry-Howard isomorphism type theory **intuitionistic logic** (law of excluded middle, LEM) **double negation** $\neg\neg p \Rightarrow p.$

$p \vee \neg p$ $p \vee \neg p$ p $\neg p$

axiom of choice $\Rightarrow$ law of excluded middle. axiom of choice

Topological interpretation

open sets Hausdorff p $\overline{p}$ open $\neg p$ p **interior** $\neg p \triangleq \overline{p}^{\circ}$. $p \cup \neg p \neq$ Universe: [*]



(2.2.1)

2.3 Higher-order logic

Propositional logic

p, q $p \wedge q, p \vee q, p \Rightarrow q, \neg p.$

[*] diagram from the book: *Classical and Non-classical Logics – an introduction to the mathematics of propositions* [Eric Schechter 2005], p.126.

First-order logic

$$\overbrace{\text{IsHuman}}^{\text{predicate}}(\overbrace{\text{John}}^{\text{object}}). \tag{2.3.1}$$

Predicate objects predicates

$$\text{Loves}(\text{John}, \text{Mary}). \tag{2.3.2}$$

First-order $\forall, \exists$ objects Mary

$$\forall x.\ \text{Loves}(x, \text{Mary}) \tag{2.3.3}$$

first-order logic predicates second-order logic.

$$\forall p.\ p(\text{Good General}) \Rightarrow p(\text{Napoleon}). \tag{2.3.4}$$

p predicates

2.4 logic with type theory

Type theory Russell **impredicative**. type theory

Curry-Howard isomorphism type theory $p, q, p \wedge q$ terms Curry-Howard $=$ types $=$ terms.

type theory (first- or higher-order) predicate logic

$$\text{IsHuman}(\text{John}) \tag{2.4.1}$$

IsHuman term (truth value) $\in \Omega = \{\top, \bot\}$. IsHuman Obj $\to \Omega$ term.

This approach leaves no room to accommodate Curry-Howard isomorphism. To do the latter, we would need Martin-Löf type theory....

2.5 Martin-Löf type theory

Curry-Howard, $A \Rightarrow B$ type:

$$\overbrace{\text{Human (Socrates)}}^{A} \Rightarrow \overbrace{\text{Mortal (Socrates)}}^{B} \tag{2.5.1}$$

$$\Omega \qquad\qquad\qquad \Omega$$

Human() Mortal() predicates type theory types. $\rightarrow$ $\Rightarrow$
Curry-Howard type theory

"simple" type theory

sum type $A + B$

product type $A \times B$

function type $A \to B$

$\vee, \wedge, \Rightarrow$. type theory

Human(Socrates) Human() Socrates arrow $\rightarrow$ arrow

Martin-Löf **type constructors**

dependent sum type Σ

dependent product type Π

Dependent sum $\sum_A B$ B depends on A. family of A $+$ product $A \times B$.

Dependent product $\prod_A B$ B depends on A. family of A $\times$ exponentiation B^A.

Dependent products can be used to define **predicates** such as Human() and Mortal(). They are of type $\mathrm{Obj} \to \Omega = \Omega^{\mathrm{Obj}} = \prod_{\mathrm{Obj}} \Omega$. *

$\sum_A B$ $\prod_A B$ types inhabited $\exists A.B(A)$ $\forall A.B(A)$.
$A \times B$ inhabited $B(A)$ B^A inhabited A send to B.

Per Martin-Löf (1942-) was the first logician to see the full importance of the connection between intuitionistic logic and type theory.

Per Martin-Löf (1942-)

2.6 Arithmetic-logic correspondence

$\wedge, \vee$ $\times, +$ fuzzy logic $\min, \max$. George Boole

$A \Rightarrow B$ B^A †

A	B	$A \Rightarrow B$	B^A
0	0	1	$0^0 = 1$
0	1	1	$1^0 = 1$
1	0	0	$0^1 = 0$
1	1	1	$1^1 = 1$

(2.6.1)

Curry-Howard correspondence $\Rightarrow$ "functional interpretation of logical deduction."

* Note that "objects" here mean logic objects, not objects in category theory.

† 0^0 1.

2.7 The problem of material implication

table (2.6.1) A B truth values types **inhabitants**. Type A inhabitant $\blacksquare$ $\varnothing$. A $= A$ type **cardinality**; The truth valuation of $A = |A|$. $A \Rightarrow B$ $|B^A|$, $\{\blacksquare\}$ $\varnothing$ $\{\blacksquare\}$ $\varnothing$ map , map $\varnothing \mapsto \{\blacksquare\}$

fuzzy logic (§7.1) strict implication $A \multimap B$ (§6.5).

material implication

$$A \wedge B \quad \vdash \quad A \Rightarrow B \tag{2.7.1}$$

$$\wedge \qquad \vdash \qquad \Rightarrow \tag{2.7.2}$$

cases. table (2.6.1) cases $A \Rightarrow B$ material implication case $A \Rightarrow B$.

AI

inductive learning of logic rules

$$\text{generators} \vdash\!\xrightarrow{\text{generate}} \text{data of the world} \tag{2.7.3}$$

generators generators of ideals, groups, function fields,

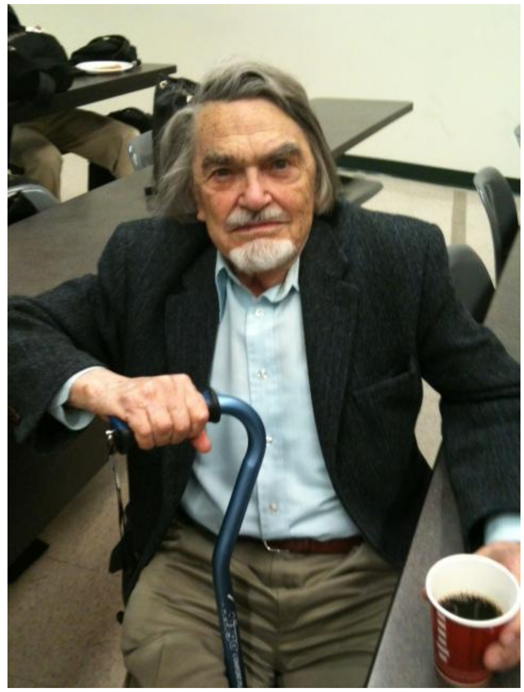
Machine learning generators $\vdash$.

learning cases $A \Rightarrow B$

3 Topos theory

4 Basics

Joachim Lambek (1922-2014)



type theory category theory Lambek Curry-Howard-Lambek

$$\begin{tikzcd} \text{logic} \arrow[rr, dash] \arrow[dr, dash] & & \text{programming} \arrow[dl, dash] \\ & \text{category theory} & \end{tikzcd} \tag{4.0.1}$$

Topos **sub-objects**.

topos $\mathcal{C}$ sub-object classifer Ω $X \to \Omega \cong$ sub-objects of X. X $X \to \Omega$ **represent**.

Set topos Ω $\{\top, \bot\}$. $X \to \Omega$ X $X \xrightarrow{\text{mathematician}} \Omega$

Topos theory commutative diagram

$$\begin{tikzcd} X \arrow[r, "!"] \arrow[d, tail, "m"'] & 1 \arrow[d, "\text{true}"] \\ Y \arrow[r, "\chi_m"'] & \Omega \end{tikzcd} \tag{4.0.2}$$

$X \xrightarrow{!} 1$ unique arrow X 1. 1 terminal object

$1 \xrightarrow{\text{true}} \Omega$ $\top$ $\perp$ $\top$, "true" arrow.

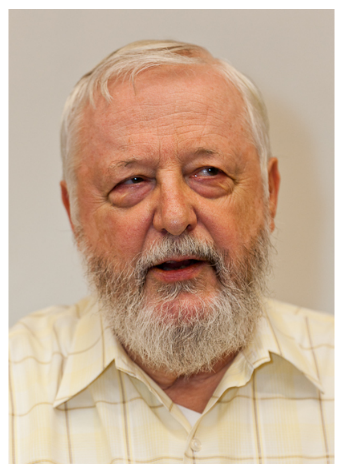
$X \overset{m}{\rightarrowtail} Y$ **monic** arrow **Set** **inclusion** map $X \overset{m}{\hookrightarrow} Y$.
X Y $X \subseteq Y$.

$Y \xrightarrow{\chi_m} \Omega$ **characteristic function**, $e \in X \subseteq Y$ $\chi(e)$
1 0. χ_m commute. χ_m $\ulcorner m \urcorner$.

Conceptual Mathematics Lawvere topos

Set diagram topos **generalize** **Set**

Topos category $P(x), \forall x, \exists x,$ Lawvere

William Lawvere (1937-) 

(4.0.2):

$$\begin{tikzcd} X \arrow[r, "!"] \arrow[d, tail, "m"'] \arrow[dr, phantom, "\lrcorner", very near start] & 1 \arrow[d, "\text{true}"] \\ Y \arrow[r, "\ulcorner m \urcorner"'] & \Omega \end{tikzcd} \tag{4.0.3}$$

$\begin{smallmatrix}1\\ \downarrow\\ \Omega\end{smallmatrix}$ **generic subobject**. **pull back** square $\lrcorner$ $\begin{smallmatrix}X\\ \downarrow\\ Y\end{smallmatrix}$ sub-object.
We say that the **property** of being a sub-object is **stable under pullbacks**.

4.1 The idea of classifying spaces

Topos $\quad Y \to \Omega \qquad$ **classifying space** **moduli space**

A moduli space is a space whose points can be put in a 1-to-1 correspondence with the objects we are interested in.

A moduli space is a manifold, or variety, which parametrises some class of geometric objects. For example:

$$\mathcal{M} = \left\{ \begin{array}{l} \text{equivalence classes of objects such as} \\ \text{Reimann surfaces, algebraic curves, etc} \end{array} \right\}. \tag{4.1.1}$$

Every family over B is the pullback of $\mathcal{C}$ via a unique map from B to $\mathcal{M}$:

$$\begin{tikzcd} \mathcal{D} \arrow[r] \arrow[d, "\phi"'] & \mathcal{C} \arrow[d, "1"] \\ B \arrow[r, "\chi"'] & \mathcal{M} \end{tikzcd} \tag{4.1.2}$$

Classifying space for principal G-bundles:

$$\begin{tikzcd} Y \arrow[r] \arrow[d, "\gamma"'] & EG \arrow[d, "\pi"] \\ Z \arrow[r, "\phi"'] & BG \end{tikzcd} \tag{4.1.3}$$

4.2 $\forall$ and $\exists$ as adjunctions

Let $\text{Forms}(\bar{X})$ denote the set of formulas with only the variables $\bar{X}$ free. ($\bar{X}$ may contain multiple variables.)

Then one can always trivially add an additional **dummy** variable Y:

$$\delta : \text{Forms}(\bar{X}) \to \text{Forms}(\bar{X}, Y) \tag{4.2.1}$$

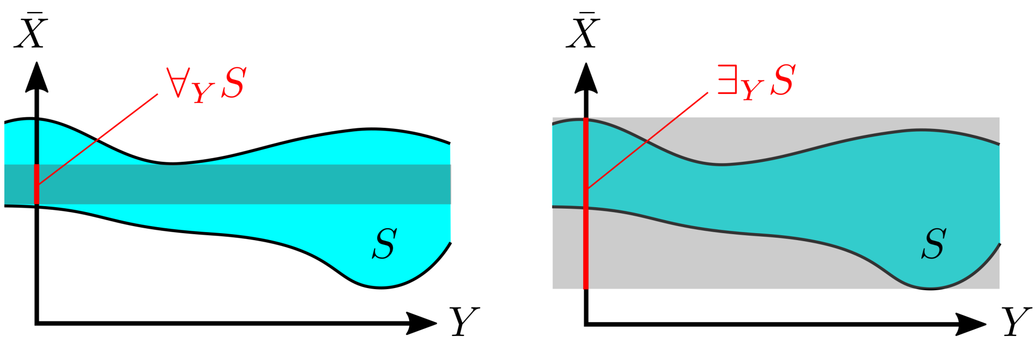
taking each formula $\Phi(\bar{X})$ to itself.

It turns out that $\exists$ and $\forall$ are **adjoints** to the map δ:

$$\begin{tikzcd} \mathrm{Forms}(\bar{X}) \arrow[r, "\delta"] & \mathrm{Forms}(\bar{X}, Y) \arrow[l, shift right=3ex, "\exists_Y"'] \arrow[l, shift left=3ex, "\forall_Y"] \end{tikzcd} \tag{4.2.2}$$

or simply denoted as $\exists \dashv \delta \dashv \forall$. This makes a lot of sense, because a formula $\Phi(\bar{X}, Y)$, after being quantified as $\forall Y.\ \Phi(\bar{X}, Y)$, turns into a formula that is **independent** of Y.

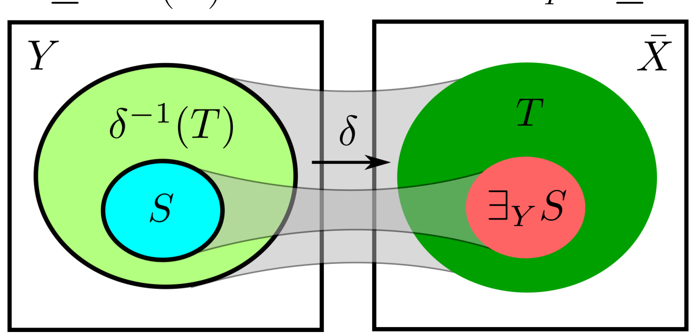
In **cylindric algebra**, the quantifiers $\forall_Y$ and $\exists_Y$ can be interpreted as **projections** where Y is the component that is "killed" by the projections:



(4.2.3)

Another way is to define with the help of a variable set T [*] (Note that $\bar{X}$ and Y are show as 2D domains below) :

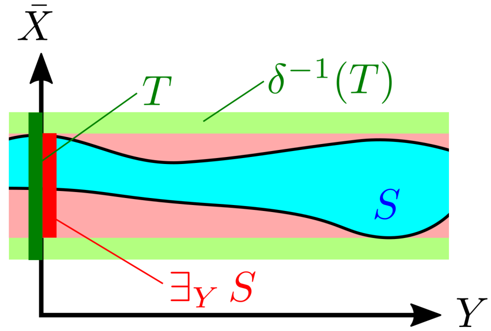
$$\forall T \subseteq \bar{X} : \qquad S \subseteq \delta^{-1}(T) \quad \Longleftrightarrow \quad \exists_Y S \subseteq T \tag{4.2.4}$$



Readers familiar with Galois theory may understand adjunctions as a generalization of **Galois connections**. In everyday language: We know S, and would like to define $\exists S$. S lives in the domain Y, $\exists S$ in the domain $\bar{X}$. We seek the help of a "shadow" T in domain $\bar{X}$ to get ahold of $\exists S$, and the shadow's "original" in domain

[*] This formula is from [1]

Y would be holding S. The following is a special case where δ is a projection, one that "kills" the dimension Y:

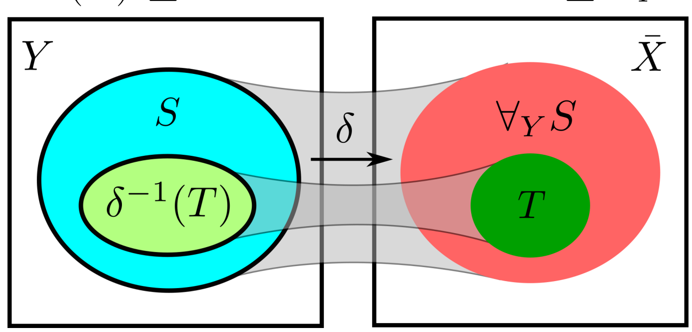


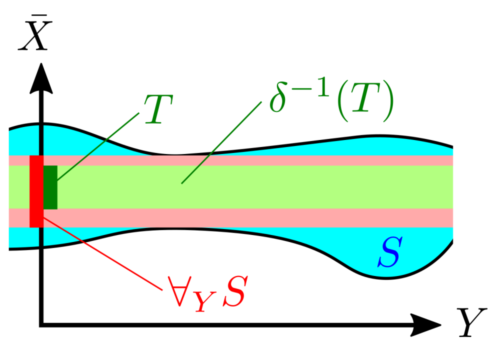
(4.2.5)

Note that in (4.2.5), δ is a projection from $\bar{X} \times Y \to X$, but the δ in (4.2.4) can be **any** map $Y \to X$, and this seems to be the most general definition of $\forall$ and $\exists$. The advantage of using categorical definitions is that they can be easily transferred to other categories, such as Hilbert space.

Similarly we have the definition of $\forall$:

$$\forall T \subseteq \bar{X}: \qquad \delta^{-1}(T) \subseteq S \quad \Longleftrightarrow \quad T \subseteq \forall_Y S \tag{4.2.6}$$





(4.2.7)

4.3 $\wedge$ and $\Rightarrow$ as product-hom adjunction

$$\begin{array}{ccc} A & & A \\ B & \vdash & B \\ C & & C \end{array} \tag{4.3.1}$$

$A \Rightarrow D$ D

$$\begin{array}{ccc} A & & A \\ B & \vdash & B \\ C & & C \\ \color{red}{A \Rightarrow D} & & D \end{array} \tag{4.3.2}$$

$\Rightarrow$ inference

$\vdash$ $\Rightarrow$ $\vdash$ (meta-logic) $\Rightarrow$

(4.3.2) $\color{red}{A \Rightarrow D}$ $A \vdash D$

$\Delta \Rightarrow \Gamma$ $\vdash$ Δ (a restriction of the $\vdash$ map to the domain Δ).
map map defining characteristic.

topos **product-hom adjunction**, $A \wedge B$ $A \Rightarrow B$

$$(A \times B) \to C \quad \simeq \quad A \to (B \to C) \tag{4.3.3}$$

$A \Rightarrow B$ $A \vdash B$. $\vdash$ AI (externalize)

4.4 Classifying topos $\leftrightharpoons$ internal language

witness propositions true "intension" proof proof objects.

proof object syntactic map **domain** map evidence?

objects morphisms

Lambek

types $\leftrightsquigarrow$ objects

terms $\leftrightsquigarrow$ morphisms

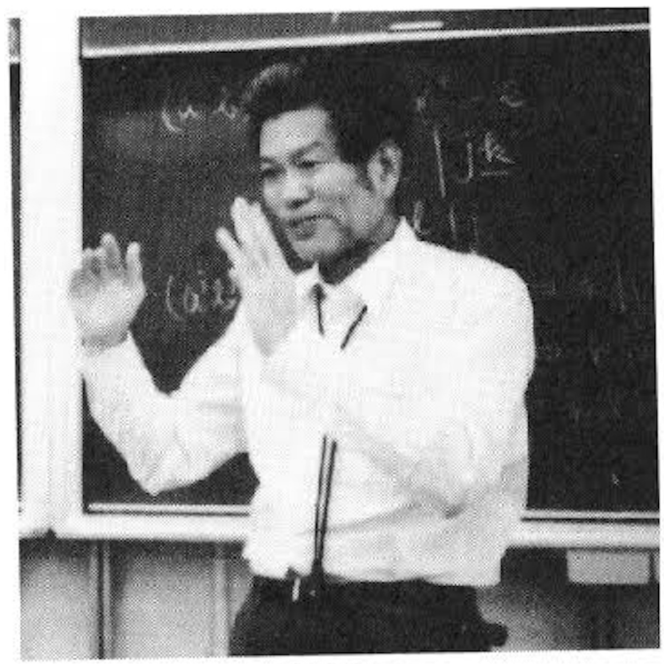
We have the following transformations betweeen two formalisms:

$$\boxed{\text{topos}}\ \mathcal{C} \underset{\text{classifying topos}}{\overset{\text{internal language}}{\rightleftarrows}} T\ \boxed{\text{type theory}}\ . \tag{4.4.1}$$

In other words,

$$\mathcal{C} = \mathcal{C}\ell(T), \quad T = \mathrm{Th}(\mathcal{C}). \tag{4.4.2}$$

4.5 Yoneda lemma



Nobuo Yoneda (1930-1996)

$\mathcal{C}$ A morphism, $A \to \bullet$. A

Set $1 \to X$ 1

$\mathbb{R} \to X$ X $\mathbb{R}$

ordered set $(\mathbb{R}, \leq)$ $0 \to x$ x **positive**.

$\bullet \to A$ A

Set $X \to 2$ 2 X , $\wp(X)$.

Top 2 open set closed set $X \to 2$ X $\mathrm{Opens}(X)$.

A functor $X : \mathcal{A} \to \mathbf{Set}$ is **representable** if $X \cong H^A$ for some $A \in \mathcal{A}$.

$H^A \qquad \mathcal{A}(A, -)$,

A **representation of X** is a choice of an object $A \in \mathcal{A}$ and an isomorphism between H^A and X.

Yoneda embedding of $\mathcal{A}$:

$$H_\bullet : \mathcal{A} \to [\mathcal{A}^{\text{op}}, \mathbf{Set}] \tag{4.5.1}$$

$$\begin{aligned} \text{For each } A \in \mathcal{A}, \text{ we have a functor} &\quad \mathcal{A} \xrightarrow{H^A} \mathbf{Set} \\ \text{Putting them all together gives a functor} &\quad \mathcal{A}^{\text{op}} \xrightarrow{H^\bullet} [\mathcal{A}, \mathbf{Set}] \\ \text{For each } A \in \mathcal{A}, \text{ we have a functor} &\quad \mathcal{A}^{\text{op}} \xrightarrow{H_A} \mathbf{Set} \\ \text{Putting them all together gives a functor} &\quad \mathcal{A} \xrightarrow{H_\bullet} [\mathcal{A}^{\text{op}}, \mathbf{Set}] \end{aligned} \tag{4.5.2}$$

$$\begin{tikzcd} \mathcal{A}^{\text{op}} \arrow[r, bend left, "H_A"] \arrow[r, bend right, "X"'] \arrow[r, phantom, "\Downarrow"] & \mathbf{Set} \end{tikzcd} \tag{4.5.3}$$

Yoneda lemma:

$$[\mathcal{A}^{\text{op}}, \mathbf{Set}](H_A, X) \cong X(A) \tag{4.5.4}$$

How sheaves gives rise to representables.

Application: symmetric neural networks

Yoneda lemma

The **Kolmogorov–Arnold representation theorem** states that every multivariate continuous function can be represented as a sum of continuous functions of one variable:

$$f(x_1, ..., x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right) \tag{4.5.5}$$

It can be specialized to such that every symmetric multivariate function can be represented as a sum of (the same) functions of one variable:

$$f(x_1, ..., x_n) = g(h(x_1) + ... + h(x_n)) \tag{4.5.6}$$

Cayley's theorem: Any group can be represented as a sub-group of a special group, namely the permutation group.

Any symmetric function can be represented as a sub-function of a special symmetric function, namely the sum.

4.6 Model theory, functorial semantics

We interpret formulas in a topos $\mathcal{E}$ by assigning each an **extension**. This is called **internal** semantics.

$$a \cdot b \longmapsto [\![a]\!] \cdot [\![b]\!] \tag{4.6.1}$$

4.7 Generalized elements

ϕ $A \xrightarrow{\phi} \Omega$ $\Omega = \{\top, \bot\}$.

$\phi(x)$ x A In category theory, we use the terminal object 1 to "pick out" elements of A, as follows:

$$1 \xrightarrow{x} A \xrightarrow{\phi} \Omega. \tag{4.7.1}$$

In **Set**, 1 $x : 1 \to A$ A

C 1

$$C \xrightarrow{x} A \xrightarrow{\phi} \Omega. \tag{4.7.2}$$

$x : C \to A$ A **generalized element**.

C **forces** $\phi(x)$, notation: $C \Vdash \phi(x)$. Paul Cohen Continuum Hypothesis forcing §4.10.

$\phi(x)$ is true **at stage** C. possible-world semantics

4.8 Internal vs external semantics

$\phi(\bullet)$ ϕ (domain) A. $\phi(x)$ x $x \in$ $\phi(x)$ **extension**

ϕ extension $x \in A, \phi(x)$ **interpretation**

The "internal" way to interpret type theory in a topos is where a formula ϕ in context $x_1 : A_1, ..., x_n : A_n$ is interpreted as a **subobject** of $A_1 \times ... \times A_n$.

internal semantics. extension internal

generalized elements **external semantics**

4.9 Kripke-Joyal / external semantics

External semantics describe which generalized elements satisfy each formula.

Generalized element $I \xrightarrow{a} A \xrightarrow{\phi} \Omega$, $a \Vdash \phi$.

A generalized element satisfies a formula iff it is a member of the formula's **extension**.

4.10 Cohen's method of forcing

forcing G G.

$$c = \{3 \in G, 57 \notin G, 873 \notin G\} \tag{4.10.1}$$

is a condition.

c G 100 1 $c \Vdash P$, P

AGI

Continuum hypothesis (CH):

$$2^{\aleph_0} = \aleph_1 \tag{4.10.2}$$

$[0, 1]$ $2^{\aleph_0}$

1878 Cantor CH

1900 Hilbert 23

Hilbert bug

1938 Gödel ZF + CH is consistent ZF cannot disprove CH

1963 Paul Cohen ZF cannot prove CH

"forcing"

Paul Cohen (1934-2007)

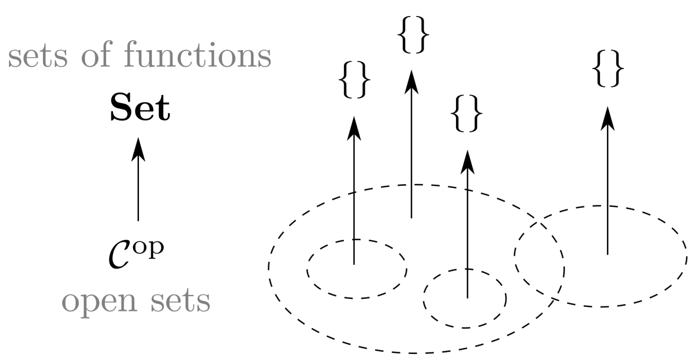
4.11 Sheaves

Sheaves capture the idea of "indexing".

Functors $\mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ are called **pre-sheaves** on $\mathcal{C}$.

Pre-sheaf sheaf sheaf condition gluing condition, $U_i \cap U_j$ presheaf sections

$\mathcal{C}^{\mathrm{op}}$ = topology of open sets



(4.11.1)

Some **Set**-valued functors are **representable**, ie, isomorphic to a hom-functor.

For an object S of a category $\mathcal{C}$, the functor

$$H^S : \mathcal{C} \to \mathbf{Set} \tag{4.11.2}$$

sends an object to its set of generalized elements of shape S. The functoriality tells us that any map $A \to B$ in $\mathcal{C}$ transforms S-elements of A into S-elements of B. For example, taking $\mathcal{C} = \mathbf{Top}$ and $S = S^1$, any continuous map $A \to B$ transforms loops in A into loops in B.

In logic, the category of predicates can be regarded as a sheaf over its domain $\begin{array}{c}\mathbf{Pred}\\ \downarrow\\ \mathbf{Set}\end{array}$.

sheaf AI

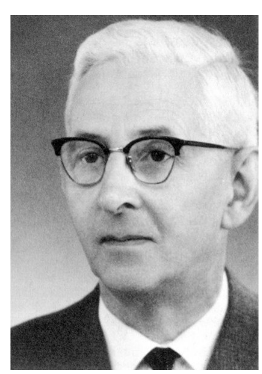
4.12 Kleene realizability

5 Intuitionistic logic

In 1933, Gödel proposed an interpretation of intuitionistic logic using possible-world semantics.

In topos theory $A \Rightarrow B$ is adjoint (via the hom-product adjunction) to $A \vdash B$, which is "okay" because it is independent of which implication (material or strict) we are using.

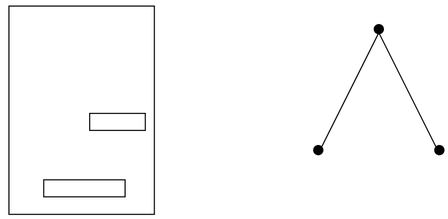
5.1 Heyting algebra

Arend Heyting (1898-1980)  (5.1.1)

1930 Heyting constructive mathematics axiomatization, called **intuitionistic logic** (IL). Heyting algebra IL Boolean algebra

Heyting algebra is to intuitionistic logic what Boolean algebra is to classical logic.

(lattice) Heyting algebra



(5.1.2)

Stone duality (every Boolean algebra is isomorphic to a topology of open sets) **Priestley**

$-\!\!\triangleright$ Heyting implication $-\!\!\triangleright$

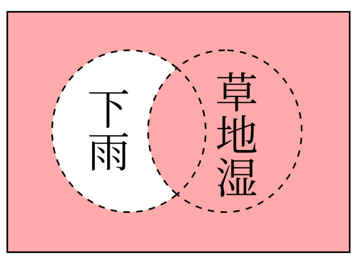
The Heyting implication $a -\!\!\triangleright b$ exists for all elements a, b, x such that:

$$x \leq (a -\!\!\triangleright b) \quad \text{iff} \quad (x \wedge a) \leq b. \tag{5.1.3}$$

Every Boolean algebra can be a Heyting algebra with the material implication defined as usual: $a \Rightarrow b \equiv \neg a \vee b$.

Heyting implication $-\!\!\triangleright$

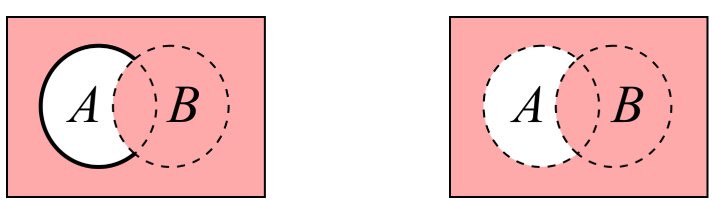
$$\forall X. \quad X \subseteq (\quad -\!\!\triangleright \quad) \quad \text{iff} \quad (X \wedge \quad) \subseteq \tag{5.1.4}$$



(5.1.5)

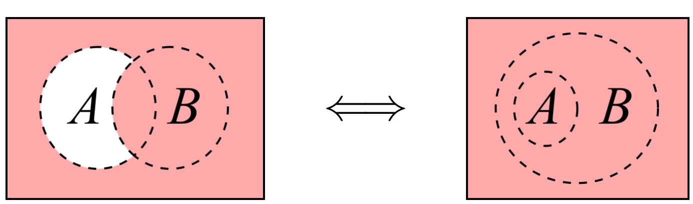
$\Rightarrow$ $-\!\!\triangleright$

classical $A \Rightarrow B$ Heyting $A -\!\!\triangleright B$



(5.1.6)

$$A \Rightarrow B \quad \Longleftrightarrow \quad A \vdash B \text{ or } A \subseteq B \tag{5.1.7}$$



(5.1.8)

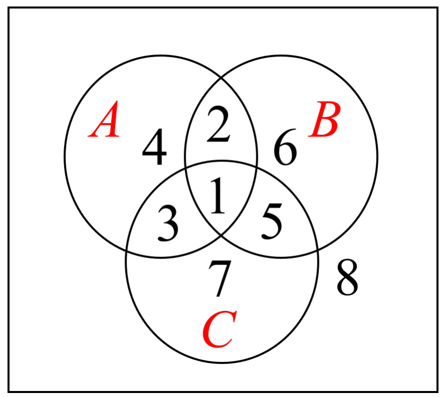
Stone duality, Boolean Heyting algebra toplogy of (open) sets

$$A \vdash B \qquad A \subseteq B \tag{5.1.9}$$

topology **truth table**

A	B	C	#
$\top$	$\top$	$\top$	1
$\top$	$\top$	$\bot$	2
$\top$	$\bot$	$\top$	3
$\top$	$\bot$	$\bot$	4
$\bot$	$\top$	$\top$	5
$\bot$	$\top$	$\bot$	6
$\bot$	$\bot$	$\top$	7
$\bot$	$\bot$	$\bot$	8

$\Longleftrightarrow$



(5.1.10)

Heyting algebra Boolean algebra

§2.7 material implication intuitionistic logic

In a topos $\mathbb{E}$, the subobject $\mathrm{Sub}_{\mathbb{E}}(A)$ is a **poset** that admits **Heyting implication**.

Using Kripke semantics, the Heyting arrow $\to$ can be defined by:

$$k \Vdash A \to B \quad \Leftrightarrow \quad \forall \ell \geq k \; (\ell \Vdash A \Rightarrow \ell \Vdash B) \tag{5.1.11}$$

Whereas the "fish-hook" **strict implication** can be defined as "A implies B necessarily":

$$A \mathrel{-\!\!3} B \quad \equiv \quad \Box(A \Rightarrow B) \tag{5.1.12}$$

The two can be regarded as equivalent via:

$$\begin{aligned} k \Vdash \Box(A \Rightarrow B) \quad &\Leftrightarrow \quad \forall \ell \geq k \; (\ell \Vdash (A \Rightarrow B)) \\ &\Leftrightarrow \quad \forall \ell \geq k \; (\ell \Vdash A \Rightarrow \ell \Vdash B) \end{aligned} \tag{5.1.13}$$

Heyting implication by possible worlds semantics machine learning classical implication

topos Heyting implication sub-objects Heyting algebra. implication arrow BHK interpretation

Heyting implication sub-object Material truth will cause an implication to exist. What does that mean?

6 Modal logic

Modalities are often conceived in terms of variation over some collection or **possible worlds**.

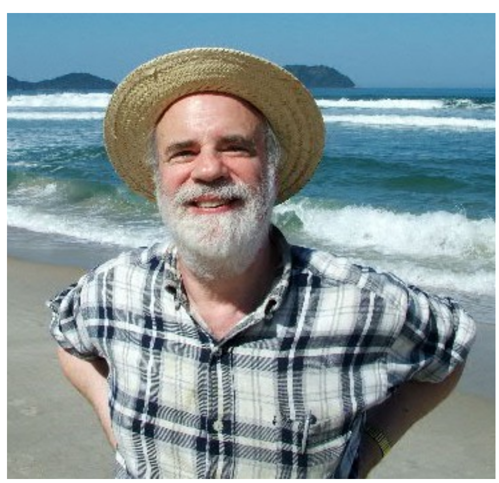
A modal operator (such as $\Box$) in the category $\mathbf{Sheaf}(X)$ is a **sheaf morphism** $\Box : \Omega \to \Omega$ satisfying 3 conditions, $\forall U \subseteq X$ and $p, q \in \Omega(U)$:

$$\begin{array}{ll} \text{a)} & p \le \Box(p) \\ \text{b)} & (\Box;\Box)(p) \le \Box(p) \\ \text{c)} & \Box(p \wedge q) = \Box(p) \wedge \Box(q) \end{array} \tag{6.0.1}$$

6.1 Possible-world semantics

Possible-world semantics is also called **intensional semantics**, as opposed to **extensional semantics** where truth values are directly assigned to propositions. Does this idea jibe with the other definition of "intension", ie, as opposed to Leibniz extensionality and also related to intensional logic?

Saul Kripke (1940-)



6.2 Computer implementation of possible worlds

modal logic frame $F = \langle W, R, D, H \rangle$,

W = set of possible worlds = $\{w_1, w_2, ...\}$

R = a relation between worlds, $w_i R w_j$

D = domain of first-order objects

$H : W \to \wp(D)$, for each world specify a subset of objects

To interpret formulas with $\Box$:

$$M \Vdash \Box A\ [w] \quad \Leftrightarrow \quad \forall w' \succeq w.\ M \Vdash A\ [w'] \tag{6.2.1}$$

quantify over all w's.

inference data

w p $\forall w.p[w]$.

summary working memory quantify.

6.3 Intensional vs extensional

"Beethoven's 9th symphony" and "Beethoven's choral symphony" has the same **extension** but different **intensions**.

6.4 Intensional logic

Possible-world semantics is also called **intensional semantics**, as opposed to **extensional semantics** where truth values are directly assigned to propositions.

Logic terms differ in intension if and only if it is **possible** for them to differ in extension. Thus, **intensional logic** interpret its terms using possible-world semantics.

6.5 Strict implication

The problem of "material implication"

Material implication $A \Rightarrow B$ $\neg A \vee B$

$$
\begin{array}{|c|c|c|}
\hline
A & B & A \Rightarrow B \\
\hline
0 & 0 & 1 \\
\hline
0 & 1 & 1 \\
\hline
1 & 0 & 0 \\
\hline
1 & 1 & 1 \\
\hline
\end{array}
\tag{6.5.1}
$$

Material implication

$$\Rightarrow \tag{6.5.2}$$

---

truth table material implication strict implication

For strict implication to make sense, it is always necessary to invoke possible-world semantics. A strict implication is always **learned** from numerous examples from experience, in accord with the philosophical tradition of "empiricism".

Strict implication is equivalent to material implication over multiple instances. The truth table of material implication agrees with the functional interpretation of implication.

7 Fuzzy logic

Lotfi Zadeh (1921-2017)

Iranian-Jewish

implication fuzzy implication material implication in Boolean algebra.

fuzzy truth value

set fuzzy proposition

fuzzy value

John
fuzzy value
John
John
imply
$\subseteq$

"Marilyn Monroe is sexy" Marilyn imply sexy subset
sexy Marilyn

Sexy(marilyn), Human(john), vs Human(Mathematicians).

What kind of mapping does this require?

7.1 Fuzzy implication

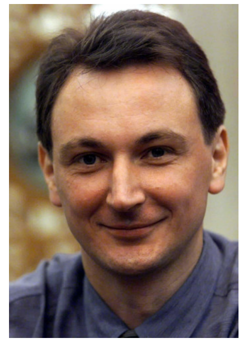
Implication generalize fuzzy logic

7.2 Fuzzy functions?

What are fuzzy functions?

8 Homotopy type theory (HoTT)

Vladimir Voevodsky (1966-2017)



HoTT types (spaces) topological homotopy homotopy type A id_A homotopy **path**.

8.1 Why HoTT may be useful

topos $A \subseteq B$ $a \in A$ subset topological topology vector space, metric space powerful

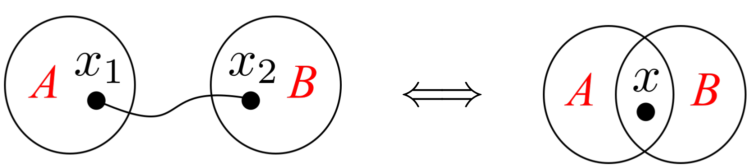
A region entity a $a \in A$ $e \in ?$ OJ Simpson $\in ?$

$a \in A$ ordered pair (a, A) abstract topology. (9.2.1) syntactic

OJ Simpson $\in$

$x \in A \wedge x \in B$ $x_1 \in A,\ x_2 \in B,\ x_1 = x_2$, and x_1, x_2 are **path-connected**. HoTT.

paths topological



(8.1.1)

$x \in A$ accidentally $x \in B$

neural network map F ordered pairs $(x \in A)$, A **proposition** can be regarded as a set of points x_i connected by paths.

8.2 HoTT levels

...	...
2	2-groupoids
1	groupoids
0	sets
-1	(mere) propositions
-2	contractable spaces

(8.2.1)

Truncation

$||A||$ is a way to obtain the **truth value** of a type A, known as **truncation**. For Voevodsky, the truth value is always binary, the type of "mere propositions". I propose that it can be generalized such that, on HoTT level 0, it provides the **fuzzy** truth value $\in [0, 1]$.

8.3 What is homotopy?

8.4 Univalence axiom

HoTT set "=" predicate, fuzzy predicate.

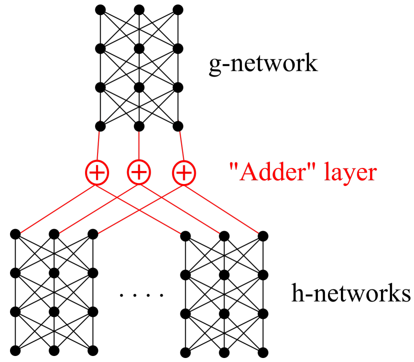
univalence axiom = fuzzy truth value binary.

9 Transfer to deep learning

logic structure (neural network, NN) impose NN "rigid" CNN (convolutional NN) convolution $f * g$ "weight sharing" NN

Symmetric NN (permutation invariant)

$$f(x_1, ..., x_n) = g(h(x_1) + ... + h(x_n)) \tag{9.0.1}$$



(9.0.2)

NN “black box” building blocks

morphisms, compositions, pullbacks, adjunctions, NN implement.

9.1 Propositional aspect

commutativity:

$$\begin{array}{rcl} A \wedge B & \equiv & B \wedge A \\ \text{it's raining} \wedge \text{lovesick} & \equiv & \text{lovesick} \wedge \text{it's raining} \end{array} \tag{9.1.1}$$

fully exploit Heyting algebra or Boolean algebra fuzzy-probabilistic logic $\wedge$ $\Rightarrow$ binary logic ....

9.2 Predicate aspect

syntax

$$\text{“Je} \bullet \text{suis} \bullet \text{étudiant”} \overset{f}{\Longrightarrow} \text{“I} \bullet \text{am} \bullet \text{student”} \tag{9.2.1}$$

words **Word2Vec** embed Curry-Howard computer implementation Curry-Howard correspondence:

$$\frac{A \Longrightarrow B}{\blacksquare \overset{f}{\longmapsto} \blacksquare} \tag{2.0.1}$$

A B f A witness $\blacksquare$ B f **domain** **co-domain** implementation (2.5.1)

$$\overbrace{\text{Human (Socrates)}}^{A} \Rightarrow \overbrace{\text{Mortal (Socrates)}}^{B} \tag{2.5.1}$$

$$\Omega \qquad\qquad\qquad \Omega$$

$\rightarrow$ **predicates** A B A Socrates Human Human(Socrates) dependent type constructor Σ

$X \in \mathcal{U}_1, P \in \mathcal{U}_2, P(X) \in \mathcal{U}_3$ $\mathcal{U}_i$ type universes ,

$$\mathcal{U}_3 = \mathcal{U}_1 \times \mathcal{U}_2 \tag{9.2.2}$$

§2.5 Σ type constructor Cartesian product

Curry-Howard syntactic ....

9.3 Implementation of topology (points and sets)

embed $\mathbb{P}$rop working-space embedding

long-term memory recall inference makes sense constants

constant predicates

absolute positions absolute position. relative position recall predicates position on-demand recall mechanism

context associative block recall.

topology LTM F F relative position inference? relative position reshape topology. reshape topology errors .....

Reshape topology generalization errors.

topology F $P(a) = 1.$ tuple (P, a). tuples to tuple map brutal syntactics. $P(a)$ type constructor $\sum_a P(a)$.

F F tuples to tuple spaces to space witness witnesses syntax.

witness, $P(a)$ positional tuple $Q(a)$ positional tuple tuple to tuple F **positional tuple to positional tuple**.

$P(a)$ (P, a) syntactic mapping

topology a a Atom predicate. constants recall recall reshape topology

constant assign **constant** constants

tuple-to-tuple, topological membership impose topological-metric **regularity** (ie, smoothness).

9.4 Modal aspect

10 Model-based AI

BERT AGI syntax $\vdash$ $\vdash$ syntactic consequence. $\models$ model-theoretic consequence.

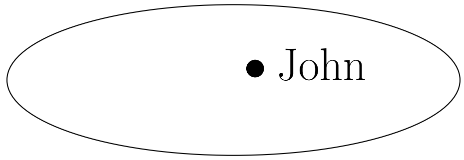
model point-set topology.

$$\begin{aligned} \text{propositions} &\Leftrightarrow \text{points} \in \text{regions or product of domains} \\ \forall \text{ formulas} &\Leftrightarrow \text{sub-regions} \end{aligned} \tag{10.0.1}$$

Model-based (logic-based) AI syntax-based working memory $P(a)$, $P = a$ $P(x)$ model-based model P extension = "**omniscient predicates**"

Then what is inference? What kind of changes would occur to the model?

Appearance of new propositions = appearance of new points / change of shape of regions.



(10.0.2)

Then how is the spatial model better than a syntactic representation (bunch of propositions)?

References

Questions, comments welcome ☺

References

[1] Abramsky and Tzevelekos. "Introduction to categories and categorical logic". In: New structures for physics. Ed. by Coecke. 2011. Chap. 1.